

BAB II

DASAR TEORI DAN TINJAUAN PUSTAKA

2.1 Dasar Teori

Dasar teori yang mendukung pembuatan aplikasi diantaranya adalah :

2.1.1 Pengertian Rental Mobil

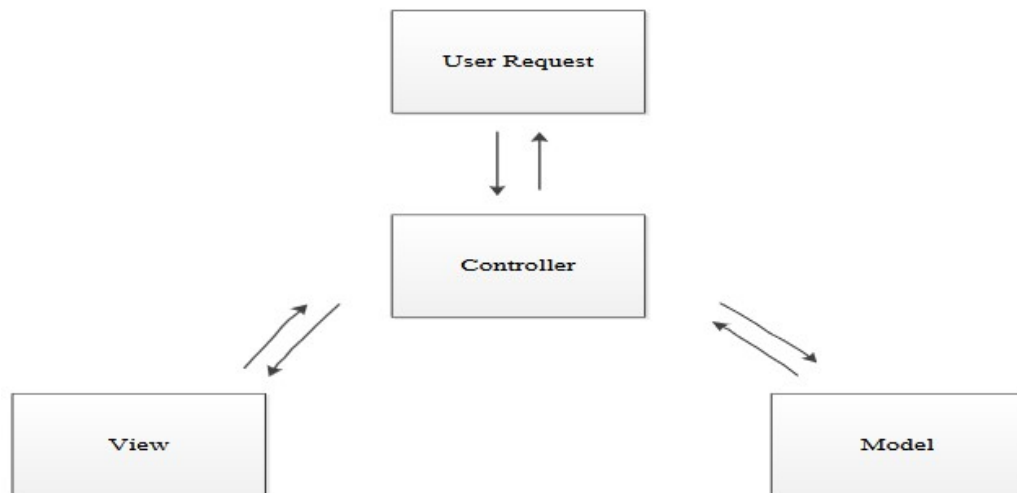
Rental Mobil adalah penyedia layanan penyewaan mobil dengan cara sewa harian ataupun kontrak dengan menggunakan driver ataupun lepas kunci, pemanfaatan rental mobil ini dapat dikembangkan sebagai terobosan bagi masyarakat atau perusahaan yang tidak memiliki alat transportasi yang akan digunakan untuk operasional.

Perjanjian sewa-menyewa (rental) mobil terdapat dua jenis, yaitu perjanjian rental mobil dengan sopir dan perjanjian rental mobil lepas kunci. Sewa mobil dengan sopir, artinya pihak yang menyewakan selain menyediakan mobil juga menyediakan sopir, sedangkan sewa mobil lepas kunci, artinya pihak yang menyewakan hanya menyediakan mobil yang disewa oleh penyewa.

2.1.2 Pengertian CodeIgniter

Famework Codeigniter adalah salah satu *framework* yang digunakan untuk membuat *website* berbasis bahasa PHP. *Framework* ini menggunakan metode MVC dalam manajemen bait program yang dibuat. Dengan *MVC (Model-View-Controller)* maka anda dapat memisahkan script yang berupa pengelolaan data ke *database*, script yang mengatur tampilan, dan script yang mengontrol semua aktifitas yang ada di website. *MVC (Model-View-Controller)* dapat membuat script tampak rapi dan berkelompok, disamping hal tersebut membuat *template* juga akan lebih mudah.

Adapun alur program aplikasi berbasis *framework* CodeIgniter dapat dilihat pada gambar 2.1



Gambar 2.1 *Model-View-Controller*

Gambar diatas menerangkan bahwa ketika datang sebuah *user request*, maka akan ditangani oleh *controller*, kemudian *controller* akan memanggil *model* jika memang diperlukan operasi database. Hasil dari *query* oleh *model* kemudian akan dikembalikan ke *controller*. Selanjutnya *controller* akan memanggil *view* yang tepat dan mengkombinasikannya dengan hasil *query model*. Hasil akhir dari operasi ini akan ditampilkan *di browser*.

Dalam konteks CodeIgniter dan aplikasi berbasis web, maka penerapan konsep MVC mengakibatkan kode program dapat dibagi menjadi tiga kategori, yaitu :

1. *Model*

Kode program (berupa *OOP class*) yang digunakan untuk memanipulasi *database*.

2. *View*

Berupa *template* html/xml atau php untuk menampilkan data pada *browser*.

3. *Controller*

Kode program (berupa *OOP class*) yang digunakan untuk mengontrol aliran aplikasi (sebagai pengontrol *model* dan *view*).

2.1.3 Konfigurasi CodeIgniter

Walaupun CodeIgniter dapat berjalan dengan konfigurasi default, tetapi untuk sebuah aplikasi yang nyata kita harus tetap melakukan konfigurasi, setidaknya pada bagian `base_url` dan router. Pengaturan `base_url` dan router sangat berguna ketika proses pengembangan aplikasi yang banyak menggunakan helper dan library.

File konfigurasi terletak dalam folder `application/config`. Adapun file-file yang terdapat dalam direktori tersebut dan sering digunakan antara lain:

1. *Config.php*. Pada file konfigurasi `config.php` berisi konfigurasi secara umum mengenai CodeIgniter, seperti peletakkan `baseurl`, `suffix`, `frontcontroller`, serta metode yang digunakan URI dan lain-lain. Adapun konfigurasi-konfigurasi yang perlu diperhatikan adalah :

- a. `$config['base_url']` - Konfigurasi ini berisi alamat url sebuah aplikasi. Jika menggunakan helper url maka konfigurasi ini harus di-set dengan benar. Contoh: aplikasi Anda akan diakses dengan menggunakan domain `www.contoh.com/app_ci` maka pada konfigurasi ini harus diisikan:

```
$config['base_url']="http://www.contoh.com/app_ci/";
```

Tetapi jika ingin menggunakan base url yang lebih fleksibel maka dapat menggantinya dengan variabel server. Contoh:

```
$config['base_url']="http://".$_SERVER['HTTP_HOST'].  
str_replace(basename($_SERVER['SCRIPT_NAME']),"",  
$_SERVER['SCRIPT_NAME']);
```

- b. `$config['index_php']` - Konfigurasi ini berisi file yang menjadi `frontcontroller`. Konfigurasi ini berhubungan dengan `base_url`, jika menggunakan `.htaccess` untuk mempercantik url maka isi variabel ini harus dikosongkan.

- c. `$config['url_suffix']` - Konfigurasi ini bertujuan untuk menambahkan akhiran pada url. Contoh mempunyai sebuah `controller page`, maka `controller` tersebut akan diakses melalui `http://localhost/index.php/page`, dengan menambahkan `url_suffix` berisi "html" maka url tadi dapat juga diakses melalui <http://localhost/index.php/page.html>.

d. `$config['language']` - Secara default CodeIgniter sudah mendukung banyak bahasa/multy language. Kita dapat mengubah pesan-pesan yang ada di dalam CodeIgniter dengan bahasa yang kita kehendaki. Untuk mengubah bahasa tersebut cukup dengan mendownload paket bahasa yang diinginkan lalu uraikan di dalam direktori `system/languages/[nama_lang]` lalu Anda tinggal mengubah `nama_lang` di `config.o$config['enable_hooks']` - Konfigurasi ini bertujuan mengaktifkan/menonaktifkan hook pada CodeIgniter. Hook dapat dikatakan event-event yang terjadi pada CodeIgniter, dimana kita bisa meletakkan fungsi di dalamnya. Hook akan bermanfaat sekali ketika Anda ingin mengubah perilaku CodeIgniter maupun untuk logging event. Contoh: Anda ingin mengubah urutan loading library dimana Anda sudah meng-extend library router untuk menggunakan database untuk aturan routing-nya. Artinya Anda harus meload library database sebelum library loader. Hal tersebut bisa dilakukan melalui hook.

e. `$config['subclass_prefix'] = 'MY_'`. Jika kita ingin mengubah/mengextend library CodeIgniter maka library tersebut harus memiliki prefik yang sama dengan konfigurasi `ini.o$config['permitted_uri_chars']`. Konfigurasi ini bertujuan untuk keamanan CodeIgniter. Konfigurasi ini menentukan karakter apa saja yang boleh digunakan di dalam uri.

f. `$config['log_threshold']`. Konfigurasi ini menentukan bagaimana sistem logging CodeIgniter bekerja. Sistem logging ini sangat membantu dalam proses pengembangan terutama ketika debugging. Jika di set 0 maka tidak ada proses logging error di CodeIgniter. Jika di set 1 maka yang dicatat hanyalah pesan-pesan kesalahan yang termasuk kesalahan PHP. Jika di set 2 maka akan menampilkan semua pesan debug dan pesan kesalahan CodeIgniter dan PHP. Jika di set 3 maka logging-logging yang berisi informasi seperti sebuah library telah di load juga akan ditampilkan. Jika di set 4 maka semuanya akan dilog mulai dari error, pesan debug sampai yang bersifat informasi.

g. `$config['log_path']`. Dikonfigurasi ini kita dapat menentukan dimana log akan diletakkan. Jika diisi kosong maka akan diletakkan di `system/logs`. Harus diingat Anda harus mengubah tingkat hak akses dari direktori tersebut menjadi dapat ditulisi / writeable jika kita menggunakan fasilitas logging.

h. Konfigurasi session. Session di CodeIgniter menggunakan cookies jadi kita dapat mengeset waktu hidup cookies/expire, nama cookies dan lain-lain melalui konfigurasi ini.

```
$config['sess_cookie_name']='Nama cookie';
$config['sess_expiration']=7200;
$config['sess_encrypt_cookie']=FALSE;
$config['sess_use_database']=FALSE;
$config['sess_table_name']='session_table';
$config['sess_match_ip']=FALSE;
$config['sess_match_useragent']=TRUE;
$config['sess_time_to_update']=300;
```

2. *Autoload.php*. Konfigurasi ini bertujuan untuk menentukan sumber daya apa yang akan diload secara otomatis. Cara penggunaannya sederhana, misalnya kita ingin meload library database, pagination dan lain-lain secara otomatis maka kita tinggal mengubah \$autoload['libraries'] menjadi :

```
$autoload['libraries']=array('database','session','pagination')
```

3. *Routes.php*. Konfigurasi di file ini bertujuan untuk menentukan kemana routing oleh library route akan dilakukan. Hal paling sederhana yang harus dilakukan adalah mengubah default controller (controller yang akan dibuka ketika tidak ada url yang diberikan oleh browser). Misalnya website kita beralamat www.koder.web.id. Maka ketika membuka website tersebut maka secara otomatis CodeIgniter akan mengalihkan ke controller default, karena tidak disertakan di dalam url kita. Adapun yang perlu diubah adalah

```
$route['default_controller']="welcome";
```

2.1.3 Desain Web Responsif

Desain web responsif adalah sebuah konsep pengembangan aplikasi web yang memungkinkan layout untuk menyesuaikan dengan resolusi layar pengguna (*viewport*). Desain ini menjadikan aplikasi web mampu menyediakan tampilan yang optimal untuk semua ukuran layar pengguna. Jadi, tidak lagi pengguna yang harus menyesuaikan perangkat yang digunakan (seperti pendekatan lawas).

Konsep desain web responsif di dalam pengembangan aplikasi web direalisasikan melalui fitur Bootstrap. Di aplikasi ini responsif yang dibuat menggunakan Bootstrap. Bootstrap sendiri merupakan library yang biasa

digunakan untuk membuat aplikasi web ataupun situs web responsive secara cepat, mudah dan gratis. Twitter Bootstrap ini terdiri dari CSS dan HTML untuk menghasilkan Grid, Layout, Typography, Table, Form, Navigation dan lain-lain. Selain itu, di dalam Bootstrap juga sudah terdapat jQuery plugins untuk menghasilkan komponen UserInterface yang cantik seperti Transitions, Modal, Dropdown, Scrollspy, Tooltip, Tab, Alert dan lain-lain. Cara penggunaan responsif menggunakan Bootstrap seperti berikut :

- Membuat form dan system grid dengan menggunakan class bootstrap di antaranya yang penting :
 - Class “form-control” pada tag <input>, <textarea> dan <select>.
 - Class “form-group” pada tag <div>.
 - Class “col-md-*” dan “col-sm”

2.2 Tinjauan Pustaka

Tinjauan pustaka ini akan menjabarkan beberapa penelitian yang terlebih dahulu dilakukan sebelum pembangunan aplikasi mobile navigasi pada persewaan mobil yang diterapkan pada proses bisnis sebuah perusahaan rental mobil. Berikut adalah beberapa penelitian lain yang menjadi acuan penulis untuk membangun aplikasi ini.

1. Penelitian dilakukan oleh (Rangga, Nurul, & Yulia, 2009) dengan judul “Pengembangan Sistem Mobile Commerce Pada Proses Rental Mobil Menggunakan Rapid Application Development”. Pada penelitian ini dilakukan pembangunan website m-commerce untuk mempermudah proses penyewaan kendaraan mobil pada suatu badan usaha, dengan menerapkan beberapa fungsionalitas diantaranya pengelolaan data mobil sampai ke transaksi pemesanan serta informasi ketersediaan mobil. Pengembangan sistem m-commerce ini menggunakan metode Rapid Application Development(RAD). Dengan demikian siklus pembangunan perangkat lunak menjadi lebih pendek atau singkat. Penerapan sistem m-commerce ini menggunakan arsitektur yang berbasis Wireless Application Protocol (WAP) sehingga website dapat diakses dari telepon seluler melalui browser.

2. Penelitian dilakukan oleh (Didik, Nurgiyatna, & Aris, 2014) dengan judul “Pembangunan Aplikasi Pemesanan Penyewaan Rental Mobil Di Prima Tour”. Dalam penelitian ini dibangun sebuah aplikasi yakni sistem informasi rental mobil yang terdapat di perusahaan Prima Tour yang menyediakan informasi mengenai transaksi untuk pemesanan mobil. Pembangunan sistem ini berbasis web, dengan bahasa pemrograman yang digunakan yakni PHP yang merupakan bahasa scripting server-side dimana pemrosesan data dilakukan pada sisi server, MySQL yang merupakan database server yang digunakan untuk penyimpanan data.

Dari kedua penelitian memiliki persamaan dengan penelitian penulis yaitu membuat aplikasi rental mobil dan penulis akan lebih mengembangkan fitur-fitur dari pemesanan mobil sebelumnya seperti penambahan cara pemesanan mobil berbasis Android menggunakan framework CodeIgniter dengan menggunakan bahasa pemrograman PHP dan database MySQL, dimana nantinya calon pelanggan dapat melakukan proses pemesanan mobil dengan mudah tidak membutuhkan waktu lama dapat di akses melalui handphone android.